

Course Syllabus

<i>Course Number/Section</i>	CS 1337.501
<i>Course Title</i>	Computer Science I
<i>Term</i>	Sprint 2013
<i>Days & Times</i>	T-TH 7:00 PM - 8:15 PM ECSN 2.112
<i>Instructor</i>	Dr. Stephen Perkins
<i>Office Location</i>	ECSS 4.609
<i>Office Phone</i>	972-883-4522 (you can call me during office hours)
<i>Email Address</i>	stephen.perkins@utdallas.edu
<i>Office Hours</i>	Tuesday and Thursday from 8:15 PM -9:15 PM in ECSN 2.112 (Other hours by appointment)
<i>TA Office Location:</i>	ECSS 2.104-A1
<i>TA</i>	Naga Anudeep Satti (Anudeep) nagaanudeepreddy.satti@utdallas.edu Mon and Wed: 10AM to 2PM ECSS 2.104-A1 Mon and Wed: 7PM to 10PM ECSS 2.104-A1

Course Pre-requisites, Co-requisites, and/or Other Restrictions

[CS 1336](#) with a grade of C or better or equivalent programming experience. (Same as CE/TE 1337) (3-0) S

Course Description

CS 1337 - Computer Science I (3 semester hours) Introduction to object-oriented software analysis, design, and development. Classes and objects. Object composition and polymorphism. Sorting, searching, recursion. Strings using core classes. Inheritance and interfaces. Graphical User Interfaces. Includes a comprehensive programming project.

Student Learning Objectives/Outcomes

After successful completion of this course, the student should have an:

- Ability to develop object oriented software solutions
 - Ability to express multi-class relationships among objects
 - Ability to implement graphical user interfaces
 - Ability to develop event driven programs
 - Ability to implement algorithms to search and sort objects
 - Ability to develop recursive programs
-

Textbooks and Materials

Required Texts

Starting Out With C++, From Control Structures through Objects, Seventh Edition, by Tony Gaddis, Addison Wesley, 2012.

We will NOT be using the MyProgramming Lab features of the book. A used version of the text book will be fine. Additional course materials, such as assignments, sample programs, and other materials will be available via eLearning site at <http://elearning.utdallas.edu>.

Course Tools

Students will be required to write programs in C++. There are a number of development environments that students may use (CodeLite, Dev C++, Eclipse, Microsoft Visual Studio Express, NetBeans, etc). For instructional purposes, this section will use the free C++ Environment for Eclipse. This is available for download here: <http://www.eclipse.org/cdt/>

Students may use any environment that will allow them to meet the submissions requirements of the course. However, if students use an environment other than eclipse, it is up to the student to make sure that their code can be compiled and run by the TA in an eclipse environment.

Help Desk

For help with issues regarding your computer, UTD maintains a walk-in help desk. Visit their Web site for details: <http://www.utdallas.edu/ir/helpdesk/>

Tutoring

For programming assistance in CS1337, a tutoring lab will be maintained. The schedule usually comes out a couple of weeks after the semester begins. Once the tutoring schedule for this semester has been released, I will make it available to you.

In addition, it is part of the TA's job to help you, so please feel free to engage with him/her at any time. And, of course, I'll be happy to help as well.

Academic Calendar

Class	Date	Class Activity	Assignment
1	Tuesday, Jan 15	Review of Syllabus	confirm access to eLearning
2	Thursday, Jan 17	Introduction to Computers C++ Fundamentals	Read Chapters 1 & 2
3	Tuesday, Jan 22	C++ Fundamentals: Decision Structures	Read Chapter 3 & 4
4	Thursday, Jan 24	C++ Fundamentals: Loops, File I/O, Functions	Read Chapter 5 & 6 Program 1 start
5	Tuesday, Jan 29	Arrays Census Day Jan 30th Last day to drop without a "W" Jan 30th	Read Chapter 7
6	Thursday, Feb 31	Arrays continued	
7	Tuesday, Feb 5	Sorting and Searching Arrays	Read Chapter 8 Program 1 due
8	Thursday, Feb 7	Text Processing	Read Chapter 10 Program 2 start
9	Tuesday, Feb 12	Exam Review	
10	Thursday, Feb 14	Problem Solving Session	
11	Tuesday, Feb 19	Exam 1 – Chapters 1-8, 10	
12	Thursday, Feb 21	Classes	Read Chapter 13
13	Tuesday, Feb 26	Classes and Objects	Program 2 due
14	Thursday, Feb 28	Classes and Objects continued	Program 3 start
15	Tuesday, Mar 5	Constructors / Destructors	Read Chapter 14
16	Thursday, Mar 7	Static Members/Friends/Copy Constructors/Overloading/this Midterm Grades Posted by Mar 9th	
	March 10 - 16	Spring break - no classes	
17	Tuesday, Mar 19	Inheritance/Derived Classes/Polymorphism	Read Chapter 15
18	Thursday, Mar 21	Standard Template Library	Program 3 Due Program 4 Start
19	Tuesday, Mar 26	Exam Review	

20	Thursday, Mar 28	Exam 2 – Chapters 13, 14, 15	
21	Tuesday, Apr 2	Exceptions	Read Chapter 16 Program 4 Due
22	Thursday, Apr 4	Function/Class Templates	Program 5 Start
23	Tuesday, Apr 9	Standard Template Library	
24	Thursday, Apr 11	Linked Lists	Read Chapter 17
25	Tuesday, Apr 16	Recursion	Read Chapter 19 Program 5 Due Program 6 start
26	Thursday, Apr 18	GUI Programming	
27	Tuesday, Apr 23	GUI Programming Continued	
28	Thursday, Apr 25	GUI Programming Continued	
29	Tuesday, Apr 30	Exam Review	Program 6 Due
30	Thursday, May 2	Exam 3 – Chapters 16, 17, 19, GUIs Last day of the class	

Please note that the instructor reserves the right to modify this calendar.

Assignments: (Subject to change, official version in ELearning)

Program #1

Purpose: Demonstrate the ability to create and execute a C++ program using moderately complex control structures and Input/Output routines.

Write a program that creates a series of one or more loan amortization tables. The program will query the user for an input file name and an output file name. The input file will contain the number of entries in the file and the information needed to generate each table (Initial Loan Principal, Annual Percentage Rate, and Monthly Payment).

The Program should open the input file and read the first entry. This entry defines the number of tables that need to be generated using the data in the file. The program should read the appropriate items for each table and then print out an amortization table including the number of Monthly Payments and the Total Interest paid for the life of each loan. This information should be printed to the screen as well as the output file.

The program should use a separate method to calculate the amortization table(s).

An example file would be:

```
3
1000.00
0.035
125.00
5000.00
0.04
165.00
200000.00
0.055
1100.00
```

Make sure you understand this Program because Program #3 will draw on this work.

Program #2

Purpose: Demonstrate the ability to create and execute a C++ program that utilizes complex data structures to perform searching and sorting.

Write a program that will perform sorting and searching across records stored in a file.

The program should first query the user for a filename.

The program should open the file and read the first entry. It defines the number of records in the file. The program should read each record and store it into memory.

It should then prompt the user for a command. There are three types of commands:

- Sort Commands print the data sorted by either (Lastname, Firstname, or Birthdate).

- Search Commands ask for a value and print any entries that contain that value.

- Quit

The input file is a text file that contains a number of records. The first line in the file contains an integer that describes how many records the file contains. After the first line, you will have sets of three lines. Each set consists of <Lastname, Firstname, Birthdate>.

An example file would be:

```
3
Smith
John
01/15/1971
Jones
Ed
06/23/1982
Burns
Sally
03/03/1984
```

Program #3

Purpose: Demonstrate the ability to create and use new classes, handle exceptions, and utilize parameterized types in C++;

Rewrite the code for Assignment 1 to utilize a typed List of "Loan" objects. You must create the Loan class. Define an *amortize* method to perform the work. If the input file contains bad values for any entry, utilize exceptions to print an error message for that loan object, ignore any more data for that loan object, and continue reading the rest of the objects.

Program #4

Purpose: Demonstrate the ability to create and use subclasses and inheritance.

Create an *Aircraft* class that has several properties that are common to all aircraft (Ex: number of engines, seat capacity).

Create three subclasses for specialized types of aircraft (Ex: Fighter, Acrobat, Commercial). Each specialized craft should have additional properties that are common to just that type of specialized craft.

Make a list of *Aircraft* and then print them utilizing their respective *toString()* methods. This method would be overridden in each subclass.

Example, to create the list you might have something like:

```
Aircraft aircraft [] = {  
    new Fighter ("F-16", "1000 lbs of smart bombs", "2 air-to-air missiles");  
    new Acrobat ("MXS", "420 degrees/sec roll rate");  
    new Commercial("Boeing-747", "UPS", "7260 Nautical Mile Range");  
}
```

Then, Loop over aircraft calling *toString()* on each object. The output should print a well formatted description of all the specific and inherited information for each craft.

Program #5

Purpose: Demonstrate the ability to utilize linked lists and recursion.

Assignment: TBD.

Program #6

Purpose: Demonstrate the ability to write a simple event driven GUI program in C++.

Assignment: TBD

Grading Policy

Projects and exams determine grades. The final grade will be composed as follows:

Programs	35%
Exams	65%

Letter grades will be assigned as follows:

98-100	A+
92-97	A
90-91	A-
88-89	B+
82-87	B
80-81	B-
78-79	C+
72-77	C
70-71	C-
68-69	D+
62-67	D
60-61	D-
Below 60	F.

All tests are closed book and closed notes. Laptop and electronic devices are NOT allowed.

There will be regularly assigned reading and homework problems. The homework problems will require the student to spend time programming a computer.

Programming assignments should be turned in by means of eLearning. Assignment files contain:

A text copy of all source code.

A text copy of your algorithm(s) and any required supporting documentation or files

Course Policies

Class Attendance

There is a strong and direct correlation between class attendance and class performance. Students who regularly attend class tend to make significantly higher final grades than those who do not. An attendance sheet will be sent around the room each class. You will be considered absent if you have not signed the attendance sheet by ten minutes after the scheduled start time of the class. Note that instructors are required to report those students who miss too many classes to the department.

Class attendance can factor in your grade (see Extra Credit)

Extra Credit

Course credit is only given for work assigned and scheduled in the course schedule. No extra work will be assigned nor will extra credit be given for any extra work performed by a student.

However, active participation in the class is worth 5%. When computing the final grade towards the end of the course, instructor may assign up to 5% additional credit based on your active role in the classroom.

Late Work

Late assignments will not be accepted. Assignments are due at 11:30 P.M. on the day listed in the syllabus.

Make-up exams

Make-up examinations will be administered **only for well-documented emergencies**. A student must make every attempt possible, via telephone and email, to notify the instructor that he/she will miss a scheduled quiz or exam prior to the scheduled date and time. See the information below for the instructor's policy regarding religious holy days.

Grade Disputes

All grade disputes must be discussed/resolved by the student with instructor within a week of posting.

Classroom Citizenship

Students are expected to be respectful to each other and to the course instructor. Disruptive behavior in the class room is not tolerated.

Each student in the class is encouraged to join/form a study group. Members of each study group should support one another in learning and understanding the course material.

Questions about Homework and other issues

All homework assignments will be graded by the TA. The instructor is responsible for grading the exams. Therefore, if you have any question at all concerning the homework assignments, please speak with the TA about it first. Even if you were to approach me first, we would still have to go back to the TA to find out what happened. It will save time to start with them first.

If for any reason you are dissatisfied with the result, however, then please come see me about the issue and we will get it straightened out. You have every right to pursue any issue that concerns you. I'm on your side and will always work with you to find a reasonable solution.

Homework Grading Criteria

All programming assignments and exams (other than the group project) are to be individual efforts. You are not to collaborate with other students. Copying of programming assignments or exams, in whole or in part, from other sources will be considered an act of scholastic dishonesty. This policy includes copying from other students, from assignments from previous semesters or from the Internet.

Programming assignments will be graded on a 100 point basis, utilizing the following criteria:

			Max Score
Source Code	Program Design	Partitioning	5%
		Organization	5%
		Efficiency	5%
		Coupling	5%
	Comments		10%
	Coding Style	Formatting	5%
		Naming	5%
		Capitalization	5%
Execution	Program Execution	No crashes	5%
		Error Recovery	5%
		Efficiency	5%
	Specification	Nominal cases	25%
		Special cases	5%
Documentation			10%
Total			100%

Keep in mind that you always want to write code that is easy to understand and is also easy to maintain.

Source Code: (45%)

Program Design: 20%

Many times you can write code that fulfills the requirements by just writing a single main() method. But most times, the problem is complicated enough to require several steps, which may be repeated one or more times. Please design your programs so that the functionality is spread across multiple methods that each accomplish a particular task, and name the method according to the task it performs.

Points Criteria

- 5 Partitioning: Is the required functionality spread logically across multiple methods (or classes)?
- 5 Organization: Is the overall program flow easy to follow? Is it easy for an outsider to figure out how your software works?
- 5 Efficiency: Do the individual methods accomplish their given tasks as efficiently as possible? Are unnecessary variables, loops, methods, and classes eliminated?
- 5 Coupling: Are methods (and classes) “loosely coupled”? Does each method only receive the data it needs in order to accomplish its task? Are the “public” methods and variables appropriately so? Is information as hidden as possible?

Comments: 10%

- Every file should have a header that includes your name, CS1337.501, & the homework number.
- Every class should have an extensive header comment explaining the purpose of the class.
- Every method should have comments explaining what it does, what its parameters are, and what values it returns.
- Significant variables and sections of code should have comments explaining their purpose. Avoid meaningless comments like “Declare the variables” or “This code adds one to the variable”.

Coding Style: 15%

You should try to follow most of the coding standards provided in the text book

Execution: (45%)

This section has to do with how well your program runs. If your program does not compile, please be aware that you may get no credit in this section.

Program Execution: 15%

Points	Criteria
5	No crashes: Does the program actually run all the way through the simplest possible test case without crashing?
5	Error Detection & Recovery: Does the program react well (does not crash) to unexpected or inconsistent events or input? Are exceptions handled appropriately?
5	Efficiency: Does the program finish executing in a reasonable amount of time?

Satisfaction of Specification: 30%

This is really the most important part: does your program do what it is supposed to do?

Points	Criteria
25	Nominal case: Does the software correctly fulfill the requirements of the assignment for the “expected” test cases?
5	Special cases: Does the software correctly handle unusual but legal test cases? Example: square root of a negative number, interest payment higher than loan payment

For programming assignments that seem arbitrarily restricted (“your program must save 4 runs”, “your program must accept up to 10 names”), points will be not taken off as long as your program meets at least those requirements. You may exceed them without penalty.

Documentation (10%)

All assignments must be submitted with documentation. At a minimum, this should include an algorithm report.

You should describe the overall flow of your program at least at a high level. If there are any parts of the program that are unusually complex, you should specify those parts in detail, using pseudocode or a flowchart.

Field Trip Policies / Off-Campus Instruction and Course Activities

No off-campus activities are scheduled.

Student Conduct & Discipline

The University of Texas System and The University of Texas at Dallas have rules and regulations for the orderly and efficient conduct of their business. It is the responsibility of each student and each student organization to be knowledgeable about the rules and regulations which govern student conduct and activities. General information on student conduct and discipline is contained in the UTD publication, *A to Z Guide*, which is provided to all registered students each academic year.

The University of Texas at Dallas administers student discipline within the procedures of recognized and established due process. Procedures are defined and described in the *Rules and Regulations, Board of Regents, The University of Texas System, Part I, Chapter VI, Section 3*, and in Title V, Rules on Student Services and Activities of the university's *Handbook of Operating Procedures*. Copies of these rules and regulations are available to students in the Office of the Dean of Students, where staff members are available to assist students in interpreting the rules and regulations (SU 1.602, 972/883-6391).

A student at the university neither loses the rights nor escapes the responsibilities of citizenship. He or she is expected to obey federal, state, and local laws as well as the Regents' Rules, university regulations, and administrative rules. Students are subject to discipline for violating the standards of conduct whether such conduct takes place on or off campus, or whether civil or criminal penalties are also imposed for such conduct.

Academic Integrity

The faculty expects from its students a high level of responsibility and academic honesty. Because the value of an academic degree depends upon the absolute integrity of the work done by the student for that degree, it is imperative that a student demonstrate a high standard of individual honor in his or her scholastic work.

Scholastic dishonesty includes, but is not limited to, statements, acts or omissions related to applications for enrollment or the award of a degree, and/or the submission as one's own work or material that is not one's own. As a general rule, scholastic dishonesty involves one of the following acts: cheating, plagiarism, collusion and/or falsifying academic records. Students suspected of academic dishonesty are subject to disciplinary proceedings.

Plagiarism, especially from the web, from portions of papers for other classes, and from any other source is unacceptable and will be dealt with under the university's policy on plagiarism (see general catalog for details). This course will use the resources of turnitin.com, which searches the web for possible plagiarism and is over 90% effective.

Email Use

The University of Texas at Dallas recognizes the value and efficiency of communication between faculty/staff and students through electronic mail. At the same time, email raises some issues concerning security and the identity of each individual in an email exchange. The university encourages all official student email correspondence be sent only to a student's U.T. Dallas email address and that faculty and staff consider email from students official only if it originates from a UTD student account. This allows the university to maintain a high degree of confidence in the identity of all individual corresponding and the security of the transmitted information. UTD furnishes each student with a free email account that is to be used in all communication with university personnel. The Department of Information Resources at U.T. Dallas provides a method for students to have their U.T. Dallas mail forwarded to other accounts.

Withdrawal from Class

The administration of this institution has set deadlines for withdrawal of any college-level courses. These dates and times are published in that semester's course catalog. Administration procedures must be followed. It is the student's responsibility to handle withdrawal requirements from any class. In other words, I cannot drop or withdraw any student. You must do the proper paperwork to ensure that you will not receive a final grade of "F" in a course if you choose not to attend the class once you are enrolled.

Student Grievance Procedures

Procedures for student grievances are found in Title V, Rules on Student Services and Activities, of the university's *Handbook of Operating Procedures*.

In attempting to resolve any student grievance regarding grades, evaluations, or other fulfillments of academic responsibility, it is the obligation of the student first to make a serious effort to resolve the matter with the instructor, supervisor, administrator, or committee with whom the grievance originates (hereafter called "the respondent"). Individual faculty members retain primary responsibility for assigning grades and evaluations. If the matter cannot be resolved at that level, the grievance must be submitted in writing to the respondent with a copy of the respondent's School Dean. If the matter is not resolved by the written response provided by the respondent, the student may submit a written appeal to the School Dean. If the grievance is not resolved by the School Dean's decision, the student may make a written appeal to the Dean of Graduate or Undergraduate Education, and the dean will appoint and convene an Academic Appeals Panel. The decision of the Academic Appeals Panel is final. The results of the academic appeals process will be distributed to all involved parties.

Copies of these rules and regulations are available to students in the Office of the Dean of Students, where staff members are available to assist students in interpreting the rules and regulations.

Incomplete Grade Policy

As per university policy, incomplete grades will be granted only for work unavoidably missed at the semester's end and only if 70% of the course work has been completed. An incomplete grade must be resolved within eight (8) weeks from the first day of the subsequent long semester. If the required work to complete the course and to remove the incomplete grade is not submitted by the specified deadline, the incomplete grade is changed automatically to a grade of F.

Disability Services

The goal of Disability Services is to provide students with disabilities educational opportunities equal to those of their non-disabled peers. Disability Services is located in room 1.610 in the Student Union. Office hours are Monday and Thursday, 8:30 a.m. to 6:30 p.m.; Tuesday and Wednesday, 8:30 a.m. to 7:30 p.m.; and Friday, 8:30 a.m. to 5:30 p.m.

The contact information for the Office of Disability Services is:
The University of Texas at Dallas, SU 22
PO Box 830688
Richardson, Texas 75083-0688
(972) 883-2098 (voice or TTY)

Essentially, the law requires that colleges and universities make those reasonable adjustments necessary to eliminate discrimination on the basis of disability. For example, it may be necessary to remove classroom prohibitions against tape recorders or animals (in the case of dog guides) for students who are blind. Occasionally an assignment requirement may be substituted (for example, a research paper versus an oral presentation for a student who is hearing impaired). Classes enrolled students with mobility impairments may have to be rescheduled in accessible facilities. The college or university may need to provide special services such as registration, note-taking, or mobility assistance.

It is the student's responsibility to notify his or her professors of the need for such an accommodation. Disability Services provides students with letters to present to faculty members to verify that the student has a disability and needs accommodations. Individuals requiring special accommodation should contact the professor after class or during office hours.

Religious Holy Days

The University of Texas at Dallas will excuse a student from class or other required activities for the travel to and observance of a religious holy day for a religion whose places of worship are exempt from property tax under Section 11.20, Tax Code, Texas Code Annotated.

The student is encouraged to notify the instructor or activity sponsor as soon as possible regarding the absence, preferably in advance of the assignment. The student, so excused, will be allowed to take the exam or complete the assignment within a reasonable time after the absence: a period equal to the length of the absence, up to a maximum of one week. A student who notifies the instructor and completes any missed exam or assignment may not be penalized for the absence. A student who fails to complete the exam or assignment within the prescribed period may receive a failing grade for that exam or assignment.

If a student or an instructor disagrees about the nature of the absence [i.e., for the purpose of observing a religious holy day] or if there is similar disagreement about whether the student has been given a reasonable time to complete any missed assignments or examinations, either the student or the instructor may request a ruling from the chief executive officer of the institution, or his or her designee. The chief executive officer or designee must take into account the legislative intent of TEC 51.911(b), and the student and instructor will abide by the decision of the chief executive officer or designee.

Off-Campus Instruction and Course Activities

Off-campus, out-of-state, and foreign instruction and activities are subject to state law and University policies and procedures regarding travel and risk-related activities. Information regarding these rules and regulations may be found at the website address given below. Additional information is available from the office of the school dean. (http://www.utdallas.edu/BusinessAffairs/Travel_Risk_Activities.htm)

These descriptions and timelines are subject to change at the discretion of the Professor.